



COURS DE TÉLÉINFORMATIQUE

SÉRIE 02

OBJECTIF PÉDAGOGIQUE :

À l'issue de cette série, vous serez capable d'identifier les différents codages des informations et le mode de transmission.

PLAN DE LA LEÇON:

INTRODUCTION

I- ÉTUDE DE L'INFORMATION À ÉCHANGE

- 1- Introduction : Nature de l'information
- 2- Codage des informations
- 3- Contrôle des erreurs de transmission
- 4- Modes de transmission

II- LES DIFFÉRENTS SYSTÈMES DE TÉLÉINFORMATIQUE

- 1- Système REMOTE BATCHE
- 2- Système TIME SHARING
- 3- Système TEMPS REEL (REEL TIME)

INTRODUCTION :

Pourquoi coder ?

Lorsque l'informatique se développa, il était nécessaire de constituer un codage adapté.

La communication nécessite la compréhension entre les deux entités communicantes. L'émetteur envoie de l'information au récepteur qui doit savoir l'interpréter pour la comprendre. Ainsi, le codage de l'information est la première étape de toute communication.

I- ÉTUDE DE L'INFORMATION À ÉCHANGE :

L'être humain mit en place des langages et créa l'écrit. Au début, l'écrit consistait essentiellement dans des dessins puis vint un alphabet plus simple à utiliser qui offrait de multiples combinaisons pour une plus grande richesse de l'expression. En réalité, les caractères de l'écrit ne sont que des symboles interprétables. L'écrit se développa et la communication par la voie écrite fut institutionnalisée en France par la création de la poste royale en 1464 par Louis XI.

1- Introduction : Nature de l'information :

En effet, la machine ne comprend que des éléments binaires (bits) : 0 et 1. Le codage consistera à combiner plusieurs bits. Ainsi, une dissociation allait se faire entre l'information et le signal généré.

Le codage de l'information s'effectue en deux étapes:

- Codage sous forme binaire (ASCII, EBCDIC, DCB...) ;
- Codage de l'élément binaire par un état physique (tension, fréquence...).

Dans le cas où deux équipements veulent communiquer en utilisant des codes différents, il est nécessaire de disposer alors de fonctions de transcodage.

2- Codage des informations:

Avec le temps et pour des raisons de compréhension humaine, nous avons développé d'autres systèmes de numérotation. Ces systèmes de numérotation font partie de la famille des systèmes de codage.

Dans cette section nous apprendrons le système de codage DCB, le Code EBCDIC, le code BAUDOT, ainsi que le code ASCII.

- DCB : Décimal Codé Binaire :

Le code DCB ressemble beaucoup au système hexadécimal. Comme celui-ci, il se compose de séries de 4 chiffres binaires.

La différence réside dans le nombre de chiffres représentés. Dans le DCB il n'y a que dix chiffres (0 à 9).

<u><i>Équivalent décimal-DCB</i></u>	
Décimal	DCB
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

- Conversion de décimal à DCB :

Pour convertir un nombre décimal en code DCB il suffit de chercher l'équivalent de chaque chiffre dans le tableau.

9 6 2
↓ ↓ ↓
1001 0110 0010

Donc $(962)_{10} = (1001\ 0110\ 0010)_{DCB}$

Au premier coup d'œil, on pourrait confondre le code DCB avec le binaire pur. Toutefois il faut faire très attention car le résultat n'est pas le même.

Si vous regardez bien l'exemple, vous pouvez constater que le nombre écrit en Code DCB et celui écrit en Binaire pure sont identiques. Toutefois, la conversion décimale n'est pas la même.

Code DCB	Binaire pure
1001 0110 0001	1001 0110 0001
↓ ↓ ↓	
9 6 1	
	$(1 \times 2^{\text{exp}11}) + (1 \times 2^{\text{exp}7}) + (1 \times 2^{\text{exp}6}) + (1 \times 2^{\text{exp}5}) + (1 \times 2^{\text{exp}0}) =$ $2048 + 256 + 64 + 32 + 1 =$
Donc (100101100001) _{DCB} = <u>961</u> ₁₀	<u>(2401)</u> ₁₀

- **ASCII (American Standard Code for Information Interchange) :**

L'apparition de l'informatique et la nécessité de disposer de codes plus riches et plus fonctionnels va mettre en évidence les limitations des codes précédents et va donner naissance à des codes contenant plus de bits. En 1963, un code à 7 bits va être développé aux Etats-Unis par l'ANSI. Ce code est connu sous le nom d'ASCII ou Alphabet International n° 5 ou Code CCITT n° 5 ou ISO 646.

Avec **7** bits, le code ASCII permet la représentation des lettres (majuscules et minuscules), des chiffres, de différents symboles (nationaux,...) et des caractères de commandes. C'est un code réellement conçu pour l'échange de données et la gestion de la communication.

Le codage des lettres et des chiffres facilite le tri et le passage de majuscule aux minuscules (et vice versa).

Des caractères ont été prévus pour :

- Les commandes de mise en page (Retour Chariot, Nouvelle Ligne...);
- Les commandes de périphériques (Device Control 1 à 4);
- Les commandes de communication et de gestion de la liaison (ACK, NAK...);

Enfin, pour munir ce code de mécanisme de détection d'erreur, un bit a été rajouté permettant ce contrôle. Ce bit est appelé bit de **parité** en raison du mécanisme mis en œuvre. C'est donc un code sur 8 bits (7+1).

Le jeu de caractères ASCII a trouvé ses limites lorsqu'on a voulu coder des symboles graphiques supplémentaires. C'est pourquoi, il fut étendu avec la norme ISO 4873.

L'usage du code ASCII (7+1) dans des protocoles de transmission pose un problème lors des transmissions de données codées ASCII 8 bits ou lors des transmissions de données non ASCII.

Exemple : Voici une petite partie du tableau ASCII

Décimal	Octal	Hex	Binaire	Caractère
000	000	00	00000000	NUL (Null char.)
001	001	01	00000001	SOH (Start of Header)
002	002	02	00000010	STX (Start of Text)
003	003	03	00000011	ETX (End of Text)
004	004	04	00000100	EOT (End of Transmission)
005	005	05	00000101	ENQ (Enquiry)
006	006	06	00000110	ACK (Acknowledgment)
007	007	07	00000111	BEL (Bell)
008	010	08	00001000	BS (Backspace)
009	011	09	00001001	HT (Horizontal Tab)
010	012	0A	00001010	LF (Line Feed)
011	013	0B	00001011	VT (Vertical Tab)
012	014	0C	00001100	FF (Form Feed)
013	015	0D	00001101	CR (Carriage Return)

- **EBCDIC (Extended Binary Coded Decimal Interchange Code) :**

Est un mode de codage des caractères sur 8 bits créé par IBM à l'époque des cartes perforées. Il existe au moins 6 versions différentes bien documentées (et de nombreuses variantes parfois créées par des concurrents d'IBM), incompatibles entre elles. Ce

INF0706/CYCLE I/SÉRIE02 INF0706. 1.2.12.2 «PROPRIÉTÉCNEDP»PAGE 5

mode de codage a été critiqué pour cette raison, mais aussi parce que certains caractères de ponctuation ne sont pas disponibles dans certaines versions. Ces disparités ont parfois été interprétées comme un moyen pour IBM de conserver ses clients captifs.

EBCDIC est encore utilisé dans les systèmes AS/400 d'IBM ainsi que sur les mainframes sous MVS, VM ou DOS/VSE.

Le succès des normes de communication entre systèmes hétérogènes (notamment TCP/IP et l'Internet, ainsi que les systèmes de cryptographie pour la transmission sécurisée et moins coûteuse via des réseaux publics) a mis fin à l'intérêt de l'EBCDIC pour tout nouveau développement, et les nombreuses versions nationales de l'EBCDIC ont également de moins en moins d'intérêt depuis l'apparition de l'UTF-EBCDIC.

• BAUDOT :

Le code Baudot est dans l'histoire un des premiers codes binaires utilisé grâce à une machine. Il est aussi appelé code télégraphique ou Alphabet International (AI) n°2 ou code CCITT n°2.

C'est un code binaire, c'est-à-dire que chaque caractère que l'on souhaite est codé par une combinaison de 0 et de 1. Le code ne prévoit que 5 bits pour coder chaque caractère, donc il n'existe que $2^5 = 32$ combinaisons. Or si on désire coder les lettres et les chiffres, il n'y a pas assez de combinaisons. C'est pourquoi le code Baudot contient deux jeux de caractères appelés Lettres (Lower Case) et Chiffres (Upper Case). En fait, l'ensemble Chiffres contient aussi d'autres symboles (ponctuation, &, #...). Deux caractères Inversion Lettres (code 31) et Inversion Chiffres (code 27) permettent de commuter entre les deux ensembles.

Évidemment, l'inconvénient réside dans des commutations fréquentes. D'autre part, ce code, bien qu'il soit plus riche que le code morse, ne traite pas les minuscules et certains symboles.

Le code Baudot est utilisé dans le réseau Télex. Il est également mis en œuvre dans certaines versions du radio télétype.

3- Le contrôle des erreurs (Error control) :

Le codage binaire est très pratique pour une utilisation dans des appareils électroniques tels qu'un ordinateur, dans lesquels l'information peut être codée grâce à la présence ou non d'un signal électrique.

Cependant, le signal électrique peut subir des perturbations (distorsion, présence de bruit), notamment lors du transport des données sur un long trajet. Ainsi, le contrôle de la validité des données est nécessaire pour certaines applications (professionnelles, bancaires, industrielles, confidentielles, relatives à la sécurité, ...).

C'est pourquoi il existe des mécanismes permettant de garantir un certain niveau d'intégrité des données, c'est-à-dire de fournir au destinataire une assurance que les données reçues sont bien similaires aux données émises. La protection contre les erreurs peut se faire de deux façons :

- Soit en fiabilisant le support de transmission, c'est-à-dire en se basant sur une protection physique. Une liaison conventionnelle a généralement un taux d'erreur compris entre 10^{-5} et 10^{-7} ;
- Soit en mettant en place des mécanismes logiques de détection et de correction des erreurs.

La plupart des systèmes de contrôle d'erreur au niveau logique sont basés sur un ajout d'information (on parle de « redondance ») permettant de vérifier la validité des données. On appelle somme de contrôle (En anglais checksum) cette information supplémentaire.

La correction d'erreurs :

Le contrôle de parité (Appelé parfois VRC, pour Vertical Redundancy Check ou Vertical Redundancy Checking) est un des systèmes de contrôle les plus simples. Il consiste à ajouter un bit supplémentaire (Appelé bit de parité) à un certain nombre de bits de données appelé **mot de code** (généralement **7 bits**, pour former un octet avec le bit de parité) dont la valeur (**0** ou **1**) est telle que le

nombre total de bits à 1 soit pair. Pour être plus explicite il consiste à ajouter un **1** si le nombre de bits du mot de code est impair, **0** dans le cas contraire.

- Prenons l'exemple suivant :

bit de parité



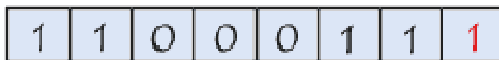
- Dans cet exemple, le nombre de bits de données à 1 est pair, le bit de parité est donc positionné à 0. Dans l'exemple suivant, par contre, les bits de données étant en nombre impair, le bit de parité est à 1 :

bit de parité



- Imaginons désormais qu'après transmission le bit de poids faible (le bit situé à droite) de l'octet précédent soit victime d'une interférence :

bit de parité



- Le bit de parité ne correspond alors plus à la parité de l'octet: Une erreur est détectée.
- Toutefois, si deux bits (ou un nombre pair de bits) venaient à se modifier simultanément lors du transport de données, aucune erreur ne serait alors détectée... .

bit de parité



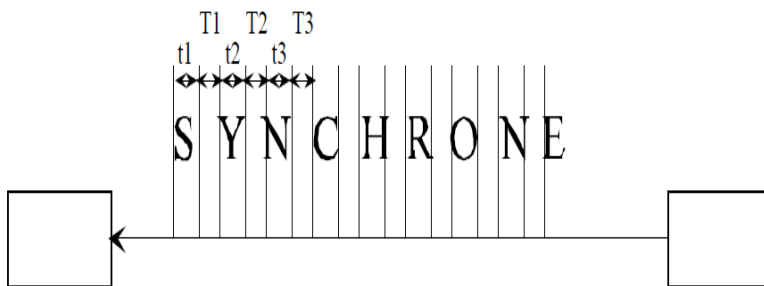
- Le système de contrôle de parité ne détectant que les erreurs en nombre impair, il ne permet donc de détecter que 50% des

erreurs. Ce système de détection d'erreurs possède également l'inconvénient majeur de ne pas permettre de corriger les erreurs détectées (le seul moyen est d'exiger la retransmission de l'octet erroné...).

4- Modes de transmission :

4.1 Transmission Synchrone :

Lors d'une transmission synchrone, le signal d'horloge de l'émetteur est transmis sur la ligne au récepteur ou reconstitué par ce dernier, ce qui évite une nouvelle synchronisation en réception et garantit des instants d'échantillonnage en phase quelle que soit la position relative du bit dans la séquence.



En pratique, l'horloge de synchronisation en réception peut être élaborée de plusieurs façons :

- Directement à partir de l'horloge d'émission si celle-ci est transmise sur une ligne séparée ;

Cas des transmissions synchrones en bande de base ou par modem sur quatre fils.

- Par reconstitution dans le modem de l'horloge d'émission à partir des instants de transition du signal de données et suivant le type de modulation.

Dans la mesure où la fréquence de l'horloge d'émission est rigoureusement égale à celle de l'horloge de réception, les débits peuvent être plus importants. De même la longueur des trames ne sont plus limitées à un caractère comme pour la transmission asynchrone mais est quelconque. En transmission synchrone, une

trame est donc composée d'un ensemble de bits pouvant être regroupés par caractères ou octets.

Il est nécessaire d'utiliser, en plus des trames de données, des trames spécifiques dites de supervision et de contrôle. Ces trames devront permettre d'établir et de libérer la liaison, de préciser l'adresse du ou des destinataires, d'acquitter ou de rejeter les trames émises, de contrôler le flux des trames d'information...

Modes de liaison :

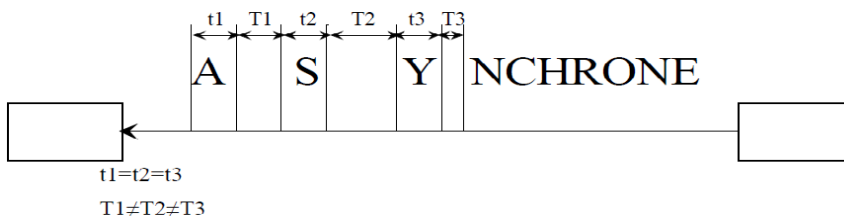
La transmission synchrone peut être associée à des liaisons point à point lorsque la voie de transmission relie deux stations, ce type de liaison est utilisé dans l'accès aux réseaux maillés comme les réseaux à grande distance (réseaux d'opérateurs X25, ATM, RNIS) ; ou en mode multipoint lorsque la voie relie plus de deux stations qui se partagent le support physique (réseaux locaux). Dans ce type de liaison, il est nécessaire de prévoir un champ adresse pour identifier les stations les unes des autres.

4.2- Transmission asynchrone :

La fonction de synchronisation sur une voie de communication a pour but d'assurer que l'information est prélevée par le récepteur aux instants où le signal est significatif.

Cette synchronisation doit s'effectuer à deux niveaux :

- Niveau bit : À quel instant le bit est-il disponible sur la ligne ?
- Niveau bloc : Instant de début et fin de bloc. Un bloc est l'unité logique à transmettre, un "message" : Par exemple un caractère ou une trame.



En émission, les données et l'horloge sont générées par l'émetteur. En réception l'horloge de synchronisation peut provenir de l'émetteur si celui-ci la transmet sur la ligne ou être interne au récepteur.

Dans le premier cas, on parle de transmission synchrone car l'émetteur et le récepteur sont synchronisés sur la même horloge de référence. Dans le deuxième cas, la transmission est dite asynchrone ou arythmique, le récepteur doit synchroniser sa propre horloge sur la séquence des bits successifs émis.

Le mode asynchrone est orienté pour une transmission par caractères, ceux-ci peuvent être émis à tout moment, la synchronisation à la réception se faisant pour chacun d'eux.

4.3- Transmission Asynchrone synchronisé : (Trameasynchrone)

Il n'y a pas de référentiel temporel permanent entre l'émetteur et le récepteur, il faut donc que l'émetteur envoie un signal de début de bloc pour indiquer au récepteur le moment où les données pourront être prélevées.

Un caractère émis sur la ligne est donc précédé d'un bit de départ (Start bit) correspondant à l'état actif (space) et à un niveau logique bas ; cette transition haut-bas va indiquer au récepteur qu'un caractère est émis et va permettre sa synchronisation. La fin de l'émission d'un caractère est indiqué par un ou plusieurs bits d'arrêt (stop bits) correspondant au niveau logique haut soit à l'état «repos» ce qui permet la distinction avec les bits de départ du caractère suivant. Cette structure est parfois nommée « START-stop ».

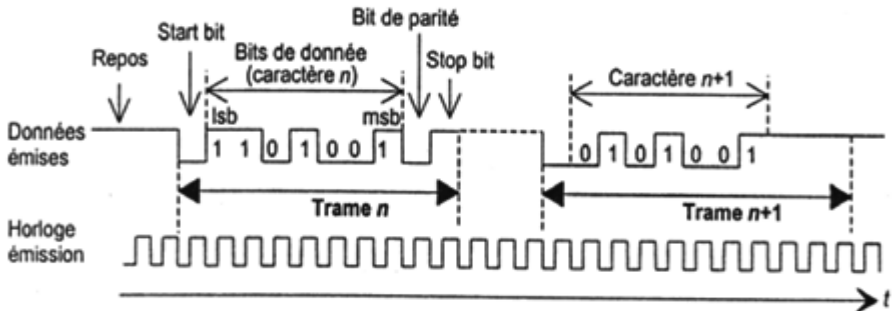
Un bit de parité, facultatif, est généré à l'émission et testé à la réception. Deux types de parité existent :

- Parité paire (**even**) : La parité est dite paire si le nombre de bits (bits de donnée et bit de parité compris) au niveau logique **1** est pair, le bit de parité est donc positionné dans l'émetteur en conséquence;
- Parité impaire (odd) : La parité est dite impaire pour un nombre impair de bits à 1. Le contrôle à la réception consiste à calculer la parité sur le caractère reçu et à la comparer à la valeur du bit

transmis par l'émetteur. Il faut donc que le choix de la parité soit le même à l'émission et à la réception.

L'état de « repos » (mark) correspond au niveau logique haut (tension négative sur le câble).

La longueur du caractère qui dépend du codage utilisé (ASCII, EBCDIC, Télex...) est généralement de 7 ou 8 bits, un certain nombre de bits sont associés à chaque caractère pour former une trame. Entre l'émission de deux trames la ligne est au repos pour une durée quelconque.



Les horloges de l'émetteur et du récepteur auront la même fréquence.

À cause de la dérive des horloges, il n'est pas possible de transmettre de longues suites Binaires (la taille des blocs est de 10 bits maximum) en effet, la durée entre chaque bit étant constante et la synchronisation se faisant sur le bit de départ, le déphasage entre l'horloge de réception et les instants correspondant aux changements de bits est d'autant plus grand que ces derniers sont éloignés du bit de départ et que la fréquence de l'horloge de réception est éloignée de celle de l'horloge d'émission. Ceci limite d'une part, le nombre de bits par trame et d'autre part, les vitesses de transmission. Vitesses de transmission courantes (en bit/s) : 75 - 110 - 150 - 300 - 600 - 1200 - 2 400 - 4800 - 9600 - 19200 - 28800 - 56600.

Il est à remarquer que ces débits ne correspondent pas aux vitesses effectives de transmission des informations dans la mesure où chaque caractère est encadré par plusieurs bits de contrôle (dans un codage ASCII sur 7 bits avec 1 bit de départ, 1 bit de parité et 1 bit de stop, 10 bits sont transmis pour 7 utiles).

II-LES DIFFÉRENTS SYSTÈMES DE TÉLÉINFORMATIQUE :

1- Système REMOTE BATCHE :

Traitement par lot ou batch permet à l'utilisateur de lancer un travail en temps différé, éventuellement à distance (**remote batch**).

Les traitements par lots sont surtout utilisés pour des tâches automatisées, notamment pour la gestion des comptes sur le parc informatique d'une entreprise, d'une université... Les travaux lancés en lots n'utilisent que les cycles processeur non utilisés par les travaux interactifs. Les lots ont donc toujours une priorité d'exécution plus basse que les interactifs mais en revanche un temps d'exécution (**time slice**) plus élevé que les interactifs de façon à rester en mémoire principale le plus longtemps possible.

2- Système TIME SHARING :

Temps partagé (time sharing) permet à chaque utilisateur de travailler de façon interactive, ce qui veut dire que chaque utilisateur peut suivre le déroulement de ses travaux par l'intermédiaire de terminaux et dialoguer avec la machine, en ayant l'impression d'en être l'unique utilisateur. Plus précisément, soient U_1, U_2, \dots, U_n les usagers du système.

L'ordinateur partageant ses ressources, soit T_i l'instant à partir duquel l'utilisateur U_i dispose du CPU durant l'intervalle de temps $d T$.

À l'instant ' T_n ', l'utilisateur U_n est pris en charge.

À l'instant **T_{n+1}** , U_1 est : Nouveau pris en charge puisque les **n** usagers ont été considérés. Le comportement du système est cyclique de période $n d T$. Si « $d T$ » est l'ordre de la milliseconde, alors les usagers seront pris en compte en n millisecondes ce qui leur donnera l'illusion de l'interactivité.

Tout ceci est possible car la période caractéristique d'un homme est de l'ordre de la seconde, alors que celle de l'ordinateur est de l'ordre de la nano seconde.

Télétraitement permet à un utilisateur parisien d'interroger par téléphone et par satellite un ordinateur à New York (utilisation de banques de données).

Multitâches (Multitasking). Le traitement multitâche permet d'exécuter simultanément plusieurs tâches différentes.

3- Système TEMPS REEL :

On parle d'un **système temps réel** lorsque ce système informatique contrôle (ou pilote) un procédé physique à une vitesse adaptée à l'évolution du procédé contrôlé.

Le traitement en temps réel assure le contrôle des opérations d'un système avec un délai de réaction très réduit (entre une micro et une milliseconde).

On peut citer l'exemple du contrôle par ordinateur de la gestion des arrivées et des départs des trains dans une gare (à la gare du Nord, aux heures de pointes, il y a trois trains qui arrivent et trois trains qui partent chaque minute. Le jeu est bien sur d'éviter les collisions....).

QUESTIONS DE COURS :

- 1- Citer le but d'utiliser la transmission asynchrone ?
- 2- Citer les deux niveaux de la transmission asynchrone ?
- 3- Définir le mode de liaison d'une transmission synchrone ?
- 4- C'est quoi une trame ?
- 5- C'est quoi un bit de parité ? citer son rôle ?

RÉPONSES :

- 1- La fonction de synchronisation a pour but d'assurer que l'information est prélevée par le récepteur aux instants.
- 2- Les deux niveaux :
 - Niveau bit : À quel instant le bit est-il disponible sur la ligne ;
 - Niveau bloc : Instant de début et fin de bloc.
- 3- Le mode de liaison d'une transmission synchrone et la liaison point à point lorsque la voie de transmission relie deux stations.
- 4- Une trame est un paquet d'information véhiculé au travers d'un support physique.
- 5- Bit de parité désigne un élément de contrôle de la qualité d'une transmission. Il est obtenu par l'analyse d'un caractère, et peut être pair ou impair selon le nombre de bits pairs présents dans le caractère.